

## **Deducing software process improvement areas from a COCOMO II-based productivity measurement**

Lotte De Rore, Monique Snoeck, Geert Poels, Guido Dedene

### **Abstract**

*At the SMEF2006 conference, we presented our experiences with the set-up of a measurement environment using the COCOMO II-model for software development projects in a company in the banking and insurance area. The set-up was part of a larger research project on managing efficiency aspects of software factory systems. One year of measurements later, a database of 22 projects is obtained. In this paper we will present our conclusions and findings after these first measurement results.*

*The effort multipliers in the COCOMO II-model represent the factors that have a linear influence on the amount of effort needed for a project. As such, they are a management instrument that gives an indication which parameters need attention within the company in order to improve the productivity. In this paper, we discuss a new kind of report we constructed in order to visualise the influence of the different effort multipliers. The goal of the report is two folded. Firstly, one can check whether the factors identified by the COCOMO II-model as effort multipliers indeed have an influence on the effort and therefore on the productivity of a project in this company. And secondly, one can check whether the amount of influence identified by the COCOMO II-model is comparable with the influence we detect in the company.*

*As we will show, even though there was only data available of one year of measurement, useful interpretations could already be given to the results as well as indications about which areas of the software development process need to be focused on in order to improve the productivity.*

### **1. Introduction**

#### **1.1. Productivity measurement project in a bank and insurance company**

With their program Expo 2005, the ICT-department of a bank and insurance company had not only the ambition to reduce the ICT-costs between 2001 and 2005 with 30%, but also to improve their ICT services and to lift up their ICT performance to a level both quantitatively and qualitatively in conformity with the market. Part of ICT activities are the development of new applications. The company rightly wonders to what extent it delivers enough value with the development of new applications given the invested time and resources. To answer this question: "What is the productivity of ICT-development in our company?" they implemented a measurement environment using the COCOMO II-model [1]. With this measurement environment, the company wants to measure the productivity of its projects on a continuous basis and wants to benchmark itself against other similar companies. Additionally, the company wants to pinpoint the different parts of the development process where improvement is possible in order to be able to adjust its development process with respect to efficiency on a continuous basis. Our paper at SMEF2006 [2] described the decisions we made and our experiences during the whole set-up process.

One year after implementing the measurement environment, it was time for a first analysis of the retrieved data. A database of 22 measured projects is obtained.

In this paper, we present a report that enabled us, although a dataset of 22 projects is rather small, to give useful interpretations to the results as well as provide indications about where actions need/can be taken in the company in order to improve the efficiency.

## 1.2. Overview of the paper

In section 2, the COCOMO II-model is briefly introduced. One of the strengths of this technique is the extensive list of cost drivers. These are the project characteristics that influence the effort needed to complete a project and are therefore a powerful instrument when searching for improvement. In section 3 we introduce a report to visualise the influence the effort multipliers had on the measured projects. We will illustrate the use of this report in the case of the bank and insurance company in section 4. With this report, we were able to indicate where improvements were advisable. Finally, in section 5, conclusions and issues for further research are given.

## 2. COCOMO II

### 2.1. COCOMO II-model

The Constructive Cost Model (COCOMO II), developed by Boehm [1], gives a formula to estimate the number of man-months it will take to develop a piece of software based on the amount of lines of code and a number of project characteristics (scale factors and effort multipliers). The effort, expressed as person-months (PM), can be calculated with the following COCOMO II-formula:

$$PM = A \times Size^E \times \prod_{i=1}^n EM_i$$

$$\text{where } E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

The size of a project is expressed in kSLOC. The parameters A and B are constant factors and the values for these two parameters were obtained by calibration of the 161 projects in the COCOMO II database and are initially equal to 2.94 and 0.91 respectively. In the exponent of the formula, one finds the scale factors (SF) that account for the economies or diseconomies of scale encountered for software projects of different sizes. The effort multipliers (EM) on the other hand are factors that have a linear influence on the effort. Effort multipliers are divided into product factors, platform factors, personnel factors and project factors. Each effort multiplier and scale factor has a range of rating levels from *very low* to *very high*. The weight assigned to the rating level of an effort multiplier indicates the amount of extra effort you need compared with a *nominal* rating level. The weight for a *nominal* rate is equal to 1. Therefore, a rating weight of 1.10 indicates that a project of this rating level will need 10% extra effort compared to the *nominal* rating. While a rating weight of 0.91 indicates that your project will need 9% less effort than in the *nominal* case. These weights were determined in the COCOMO II-model by linear regression on the 161 projects in their database.

However, projects in your own company might differ from the projects used to set the values of all these parameters. Therefore, once own project data is collected, one can perform a calibration to obtain a better predictive model adapted to the own environment.

## 2.2. COCOMO II as a model for productivity measurements

The COCOMO II-model estimates the workload a project with certain characteristics would need. Hence, we can use this estimation to benchmark the project productivity against the productivity predicted by the COCOMO II-model by comparing the estimated effort with the actual workload of the project. In other words, we use the COCOMO II-estimation as the norm for the productivity measurement. Once the formula is calibrated to the current situation in a company, in the next measurements, one will benchmark projects no longer with the COCOMO II-norm but with a company own norm.

## 2.3. Cost drivers: a source of information

One of the major strengths of the COCOMO II-model is the extensive list of cost drivers (effort multipliers and scale factors). These cost drivers capture the characteristics of the software development that affect the effort to complete the project. Hence, these factors are a very useful source of information in addition to the relative productivity of a project computed as the comparison between the actual effort and the estimated effort according to COCOMO II, to see whether your project is productive according to the COCOMO II-model. By looking at the ratings of the cost drivers for each project (no influence, positive, negative influence) the cost drivers can indicate which characteristics of your project had an influence on the estimated effort. As such, they are a management instrument that gives an indication which parameters need attention within the company in order to improve the productivity. In the remainder, we only focus on the effort multipliers and no longer on the scale factors.

As explained before, the actual value of the effect of a cost driver was determined by using the 161 projects in the COCOMO II-database. Since very little information is available about these 161 projects, we can assume that chances are real that they are different from the own projects. Therefore, it is advisable to calibrate the model to determine whether the values are correct for the own environment. In the next section we describe which reports can be produced to identify the real effect of an effort multiplier in the project data.

## 3. Reports for effort multipliers

### 3.1. Initial idea

The initial idea, as described in [2], was to construct a graph for each effort multiplier in which the relative productivity (the COCOMO II-estimated effort divided by the actual effort) was plotted against the rating of the effort multiplier. The goal with this report was to identify for which ratings of the effort multiplier most projects were productive (relative productivity  $> 1$ ) and for which ratings most projects were unproductive (relative productivity  $< 1$ ). As such, one could obtain suggestions for which ratings of the effort multipliers some attention was needed in order to improve the productivity or in order to obtain a change in the rating of the effort multiplier. An example of the kind of report we expected to obtain is shown in Figure 1(a). In this report, we set out the relative productivity of projects against the rating for a particular effort multiplier  $EM_i$ . The position of the different projects with the same rating for  $EM_i$  is presented with a line and the length of this line indicates the estimation error or the variance of the estimation. From the report shown in figure 1 (a) we see that projects with a rating of *very low* for  $EM_i$  have a lower relative productivity rating than the projects who rated  $EM_i$  as (e.g.) *low*. Hence, one can conclude it is advisable to focus the attention on project with a lower rating and develop improvement action for these projects as they have a lower relative productivity. Remark that in the case of a perfect estimation model, all the projects would be on the dashed line (relative productivity = 1).

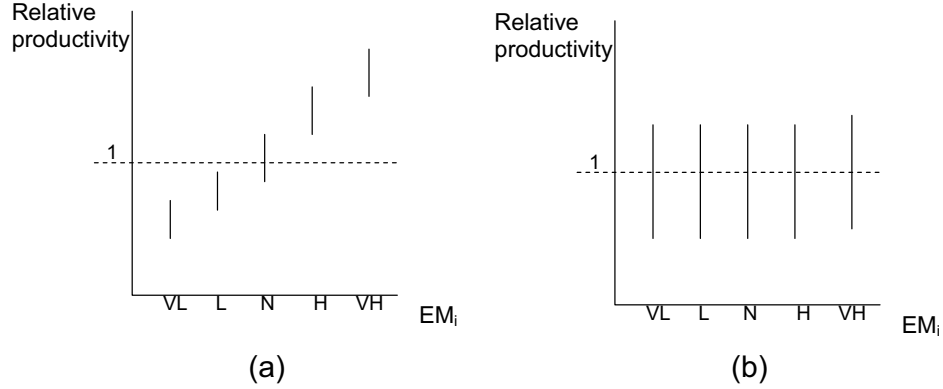


Figure 1: initial report

However, after we produced the reports with the first measurement results of our case study, we didn't feel this report provided enough information. Most reports looked more like Figure 1 (b): for each rating of an effort multiplier, we had productive projects as well as non productive projects. What we really want to see in a report is whether the effort multipliers identified in the COCOMO II-model are indeed factors that influence the effort of the projects in the company. And if an effort multiplier indeed has an influence on the effort, is this influence comparable with the effect we measure in the data we retrieved from the projects we measured. We produced a new kind of report which we present in the next section.

### 3.2. New Report: influence of the EM

Assume we focus on the effort multiplier EM<sub>i</sub>. What we want to see is whether this effort multiplier has indeed an influence on the effort of the projects in this particular company. When our estimated effort is equal to our actual effort, we get the following equation:

$$Effort = A \times Size^E \times \prod_{\substack{j=1 \\ j \neq i}}^n EM_j \times EM_i$$

In order to investigate the influence of EM<sub>i</sub>, we calculate the 'normalised effort with respect to EM<sub>i</sub>': this is the effort divided by all the other effort multipliers. As such we obtain an effort where the influences of all the other effort multipliers are neutralised. As an example, consider a project with two effort multipliers EM<sub>1</sub> and EM<sub>2</sub>, with respectively values 1.10 and 1.20. The total Effort is calculated as  $A \times Size^E \times 1.10 \times 1.20 = A \times Size^E \times 1.32$ . If we want to investigate the effect of EM<sub>1</sub> only, we need to divide the calculated Effort by EM<sub>2</sub> to eliminate its effect: Normalised Effort =  $(A \times Size^E \times 1.10 \times 1.20) / 1.20$ . Remark that we implicitly assume here that the influence of the other effort multipliers is estimated correctly by the COCOMO II-formula. So in general we can state that:

$$NormalizedEffort_{EM_i} = \frac{Effort}{\prod_{\substack{j=1 \\ j \neq i}}^n EM_j} = A \times EM_i \times Size^E$$

As we can see from this equation, there is a linear relationship between the normalised effort and a size measure (namely:  $Size^E$ ); the quantification of this linear relationship is dependent of the value of the effort multiplier. Consequently, we have expressed the influence of the effort multiplier  $EM_i$  as a linear relationship between a size measure and an effort measure.

In the report, the Normalised Effort $_{EM_i}$  is plotted against  $Size^E$ . A regression line with intercept equal to zero can be determined through the different data points with the same rating for the effort multiplier under consideration. The equation for this regression line will be

$$NormalizedEffort_{EM_i} = R \times Size^E$$

The slope of the regression line, denoted with  $R$ , is directly proportional with the value for the effort multiplier. Hence, we found a way to quantify the influence the effort multiplier has on the projects included in the analysis.

### 3.3. Interpretation of the report

The value of an effort multiplier indicates how much extra or less effort a project needs due to the rating for this particular effort multiplier. A nominal rating corresponds to no extra effort. Therefore, we will group the projects according to the rating given to the effort multiplier  $EM_i$ . For each group of projects with a given rating we will calculate a regression line and compare this line with the regression line for the projects with a *nominal* rating. In other words, as the slope of the regression line is directly proportional with the value for the effort multiplier for a particular rating, we will compare the slope of each regression line with the slope of the *nominal* regression line (this is the regression line through the projects that have a *nominal* rating for the effort multiplier). This will provide us with information about the effect of the different ratings of the effort multiplier.

For example, assume the COCOMO II-value for a high rating of the effort multiplier is equal to 1.10. This means that according to the COCOMO II-model, a project with a high rating for the effort multiplier will need 10% more effort than a project with a nominal rating for this effort multiplier. Comparing the slopes of the high regression line and the nominal regression line will provide us the information whether the effect of this effort multiplier predicted by the COCOMO II-model can also be detected in the data of the analysed projects.

When we get the same ratio (in the example 1.10), we can conclude that the effort multiplier has indeed an effect on the effort and that the amount of this effect measured in the analysed projects is comparable to the estimated effect by the COCOMO II. However, when we get a significantly different ratio, we can conclude that there is an influence but that the effect of this effort multiplier is different in this company. This indicates an area of investigation as there might be an opportunity for productivity improvement. For example, assume we get the ratio 1.20. This means for the analysed projects, a project with a *high* rating will not need 10% more effort as prescribed by the COCOMO II-model, but needs 20% more effort than a *nominal* project.

On the other hand, when we get a ratio equal to 1, we can conclude that the effort multiplier does not have an effect on the effort. And therefore, we should not credit projects with a *high* rating with more effort in the estimation.

In summary, comparing each regression line with the *nominal* regression line yields two types of information: Firstly, when the slopes are different this demonstrates that the effort multiplier has indeed an effect on the effort. Secondly, the comparison of the values also shows whether the effect is comparable to the amount prescribed by the COCOMO II-model.

## 4. Experiences in the bank and insurance company

### 4.1. First measurement results

After one year of measuring, a first analysis is performed on the retrieved data base. The data base consists of 22 projects. All these projects are bank or insurance applications. We only measured new application or added functionality. No migration projects or conversions due to technical reasons were measured.

The development environment was relative stable during the measurement period and is actually already stable for some years. However, we have to say that the administrative discipline lacks sometimes, which means that in some cases the recorded actual effort differs slightly from the real actual effort. This introduces some uncertainty in the measurement results. However, as the numbers we obtain are never used as absolute results, but only as an indication of areas that need further investigation, this is not a problem.

The size of the 22 projects is in a range from 1.2 kSLOC to 158 kSLOC. The effort ranges from 24 person weeks to 351 person weeks. The accuracy of the predictions by the COCOMO II-model is shown in Table 1. Only 5 of the 22 projects are predicted within 20% of the actual effort and 7 of the 22 projects within 30% of the actual effort. To improve the accuracy, the 22 projects were used to calibrate the two constant factors, A and B, in the COCOMO II-formula. After the calibration, the prediction for 7 of the 22 projects is within 20% and for 11 of the 22 projects within 30% of the actual effort. As can be seen in Table 2, the mean magnitude of relative error (MMRE) indicates a significant improvement in the estimation for both the projects that are estimated well (within 30% of the actual) as those that are estimated poorly (not within 30%). For the calibrated effort, the average error is around 10% for 50% of the projects (these within 30% of the actual effort) and 79% for the 50% projects not within 30% of the actual effort.

Table 1: Prediction accuracy for the 22 projects

	COCOMO II	Calibrated A and B
Pred (.20)	23%	32%
Pred (.30)	32%	50%

Table 2: MMRE before and after calibration

MMRE	COCOMO II	Calibrated A and B
Projects estimated within 30%	17%	10%
Projects estimated not within 30%	100%	79%

### 4.2. Report for the effort multipliers

As explained in the previous section, the most powerful information can be found in the cost drivers of the COCOMO II-model. This information can be revealed with the report we described in section 3.2. In the remainder of this section we will illustrate the reports we obtained for the LTEX, PLEX, DATA, STOR and TIME effort multiplier.

Because we only have 22 projects and additionally, these projects are scattered over the several ratings for each effort multiplier, we are aware that this amount of data is insufficient to provide enough confidence that the slopes capture the correct effect of the effort multiplier. Therefore we did not use the slopes as an absolute value, but we used the reports as input for a discussion with the experts in the several domains of the effort multipliers. When the slope corresponds with the feeling/experience of the experts, we certainly have an indication about the real effect of this effort multiplier in this company.

Remark that the slopes of the regression lines do not have the same value as the values for the effort multipliers. The reason is that the factor A is also included in the slope as well as the fact that within the company we measure in weeks rather than months and one month consists of 148 hours, while in the COCOMO II-model, a month consists of 152 hours. However, these factors are constant for every project, so when we compare the slope of two regression lines, these factors will be neutralised.

#### 4.2.1. Effort Multiplier: LTEX

Figure 2 shows the report for effort multiplier *Language and Tool Experience* (LTEX). Two regression lines are plotted: one for the nominal rating of LTEX and one for the *high* rating. The *high* slope in proportion to the *nominal* slope shows that a team with an overall language and tool experience of 3 years with the languages and tools used in the project (*high* rating) will only need 86% of the effort compared with a team with an overall experience of 1 year (*nominal* rating). This is a slightly larger effort reduction than COCOMO II predicts (91%).

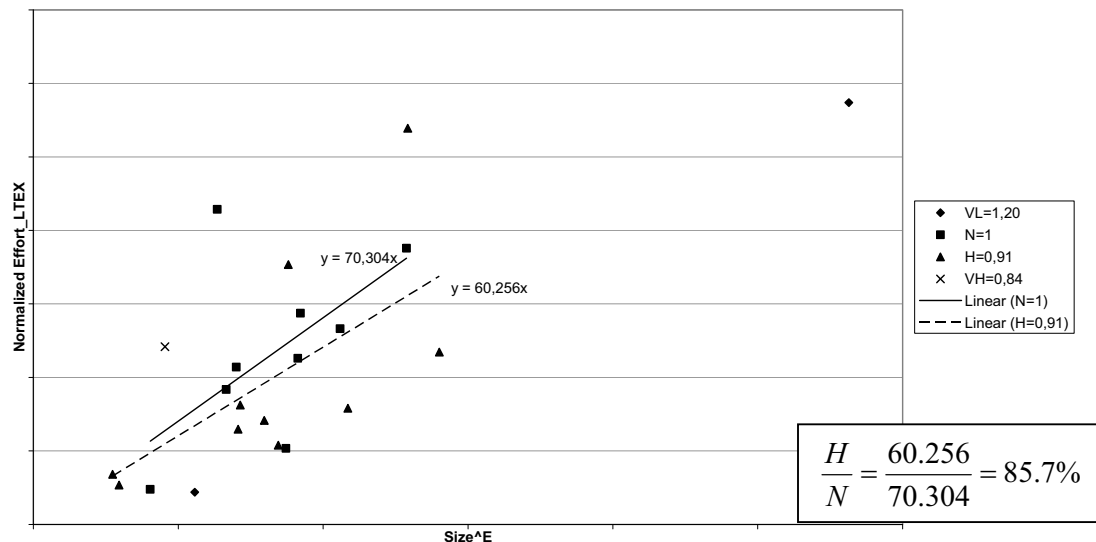


Figure 2: LTEX-report

As the effort reduction with more language and tool experienced teams is slightly larger than according to the COCOMO II-prediction, it might be advisable to aim for a higher rating for this effort multiplier. This means that we should strive to put together teams with a larger overall language and team experience. However, the company deals with a technology switch at this moment and in addition it employs a lot of external people who are not familiar with these technologies and therefore need education. With respect to this cost driver, at this moment, there is no action for productivity improvement possible.

#### 4.2.2. Effort Multiplier: PLEX

Figure 3 shows the report for effort multiplier *Platform Experience* (PLEX). Two regression lines are plotted: one for the *high* rating of PLEX (i.e. an average of 3 years experience with the platform) and one for the *very high* rating (i.e. an average of 6 years of experience with the platform). There is not enough data to plot a *nominal* regression line to compare our slopes with. However, it is clear that the slopes of the high and very high rate are equal, which means that a rating of *high* and of *very high* have the same influence on the effort. COCOMO II, on the other hand, predicts an additional effort reduction of 6% for projects with a rating of *very high* for PLEX compared to projects with a rating of *high* for this effort multiplier.

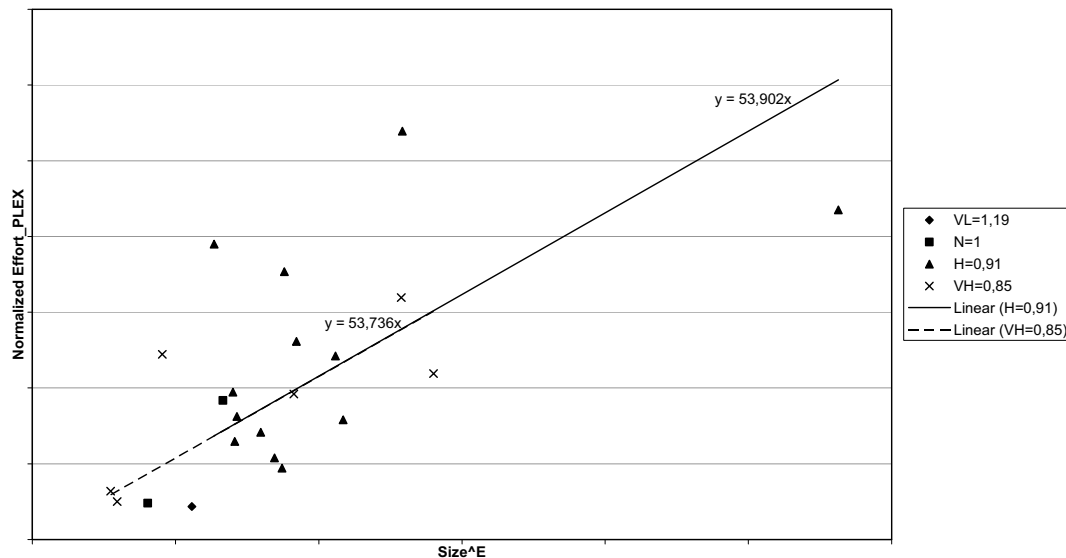


Figure 3: PLEX-report

As the equality of the slopes of the regression lines indicates, the platform experience has no further influence on the effort and productivity within this company once the team reaches an average of three years of experience. It seems plausible that after 3 years of experience with the same platform you know everything about this platform. However, as there is not enough data available about the less experienced teams, we can not generalise this observation and state that this effort multiplier has no influence at all in the company. Most likely, also in this company, there will be an effort reduction compared with projects with a nominal or lower rating. Nevertheless, teams with an average experience of more than 3 years should not have a further reduction of the allowed development time than teams with an average experience of three years.

#### 4.2.3. Effort Multiplier: DATA

Figure 4 shows the report for effort multiplier *Database size* (DATA). This effort multiplier captures the effect of large test data requirements on product development. Three regression lines are plotted: one for the *low* rating, one for the *nominal* rating and one for the *high* rating. Comparing the slopes shows that this effort multiplier has more effect than predicted by COCOMO II. A *low* rating (use a copy of production data in the accept environment as test data) will lead to a reduction of 37% in the effort, while a *high* rating (create considerable amount of own test data) will lead to an increase of 23% in the effort



compared to the *nominal* rating (create a minimum of own test data). These in- and decreases are larger than predicted in the COCOMO II-model (10% decrease for *low* rating and 14% increase for *high* rating).

Although we can not take the slopes as absolute numbers, from this report we can state that the DATA effort multiplier has a serious effect on projects in this company and therefore definitely needs attention. An inquiry with the testing team revealed that there are indeed some issues with test data, but they had no idea about the amount of effect on projects. A possible explanation for the values is that the required effort to find and manage test data is substantial in this company. Additionally, everyone works on the same test data from the infrastructure systems. This means, once a project has used this test data, afterwards most probably many updates have been made to the data by other projects. As a result, the original test data is lost and therefore it is no longer usable for the own project. For example, suppose you need to test software on account numbers and you need accounts with a negative balance. When an account has initially a negative balance, it is possible that after the test run this account has a positive balance and therefore it is no longer useful. The problem is that after one or more test runs, one does not know anymore whether the balance of the accounts in the test data set is positive or negative. This leads to additional effort required to re-create a usable set of test data over and over again.

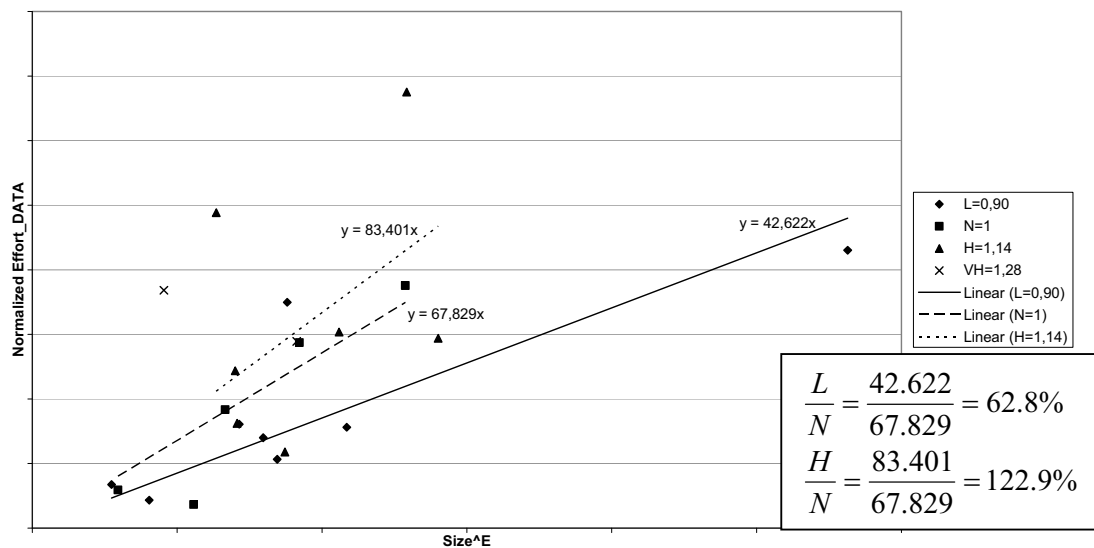


Figure 4: DATA-report

Because of the identified influence of this test data factor, the company realised it was necessary to undertake action to improve the test process. The report provided enough convincing material to build a business case in order to receive the permission from the CIO to spend the necessary resources to facilitate the retrieval and creation process of physical test data. The test team identified three potential actions: firstly, the construction of master and work test data bases. The master test data base is a stable set of test data that can be copied to a work data base for the test cases in projects. This should resolve the problems connected with the fact that everyone uses the same test data: the master data set always allows returning to the initial (and known) status of test data. Secondly, a set of standard queries on infrastructure systems can be supplied to more easily retrieve test cases. Thirdly, test data creation scripts can be provided to (re)initialise a set of test data.

This third option is an alternative for the first one that can be very expensive if it leads to an extra test environment. However, this third option might have the restriction that the scripts can only be applied to relative simple structures of physical data. These three options will be further investigated to decide which one can be implemented.

#### 4.2.4. Effort multipliers: STOR and TIME

Figure 5 and Figure 6 show the reports for effort multiplier *Main Storage Constraint* (STOR) and *Execution Time Constraint* (TIME). In the company, they don't make a difference between these two parameters and therefore the two measures are considered as a measurement for taking into account performance issues. For the STOR effort multiplier (Figure 5), a *high* rating regression line and a *nominal* rating regression line are plotted. The ratio between the slopes indicate that a project with a *high* rating (some extra attention for main storage issues) and *nominal* rating (no extra attention for main storage issues) leads to an increase of 73% in the effort, which is a large difference with the 5% predicted by the COCOMO II-model.

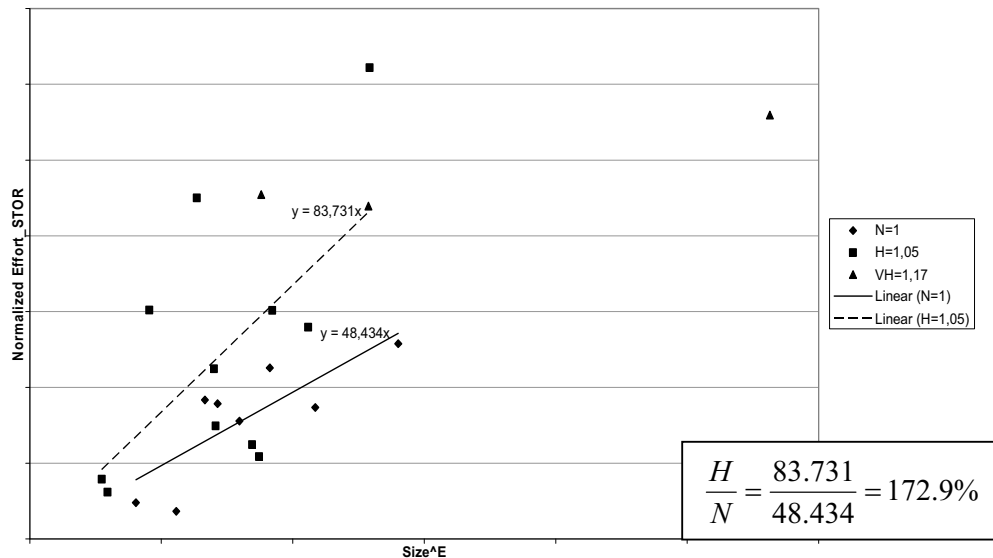


Figure 5: STOR-report

For the TIME effort multiplier (Figure 6), three regression lines are plotted (*nominal*, *high* and *very high*). Similarly as with the STOR effort multiplier, we see a large increase in effort when extra attention is needed for execution time issues. A *high* rating leads to an increase of 97% and a *very high* rating leads to an increase of 120% in the effort compared with the nominal rating. The COCOMO II-model predicts an increase of 11% and 29% for respectively a *high* and *very high* rating.

According to the experts, the values we retrieve with these reports are higher than what they experience in practice, although they agree that taking into account limitations with respect to main storage and time execution means extra design work for the project team. Nevertheless, these reports are definitely an indication that dealing with performance issues needs improved assistance in this company. At the time the measurements took place, there was not enough knowledge about performance issues present. Also project teams would spend too little attention to performance issues during the functional design stage, which would lead to rework in further stage in the development cycle. Furthermore, there was no procedure available for the set up of online performance tests.

And finally, the complexity of systems and transactions constantly increases. All these factors can give an explanation why performance is an issue that needs attention.

Currently, there is already ongoing work, mainly through coaching, to improve the knowledge about performance in project teams. As it is identified that performance issues are not tackled from the beginning of the development cycle, a new initiative has been taken to have more attention for performance issues during the project inception phase, more particularly at the moment the requirements of the physical architecture are defined.

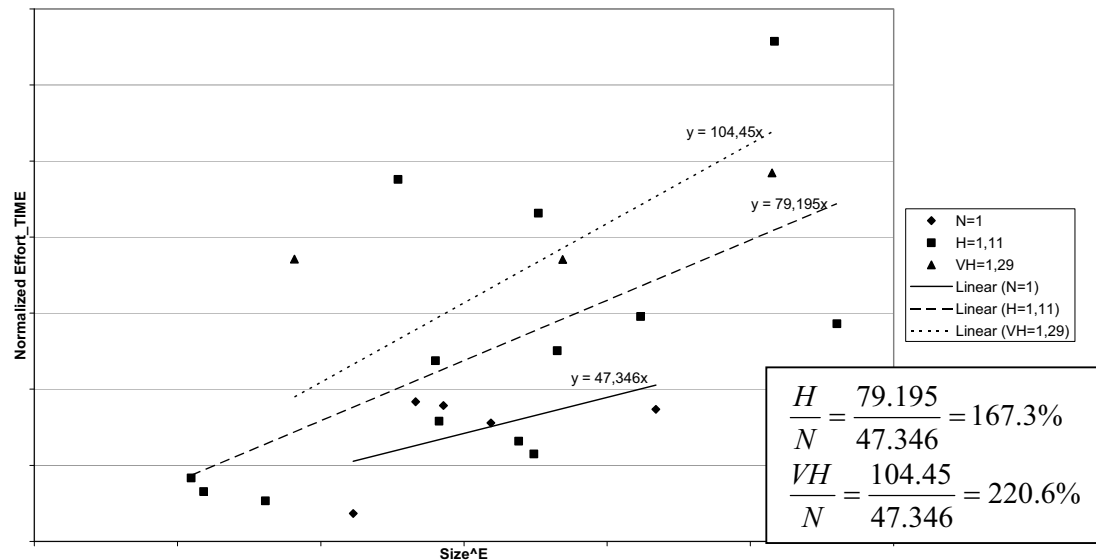


Figure 6: TIME-report

## 5. Conclusions and Future research

As we can see from our case study in the bank and insurance company, the proposed report where we use the slope of the regression line between  $Size^E$  and the Normalised Effort can provide a lot of information even when you only have a rather small amount of project data. However, these reports do not identify the actions you can undertake to improve the productivity. Yet they are an instrument to identify the places where possible problems occur and consequently where improvement is possible. When the results of this report correspond with the feeling and experience of the experts, this is a clear indication that there is an opportunity for productivity improvement. The report can be used to build a business case and to visualise to management that it is indeed of importance to invest in software process improvement measures to reduce the (negative) effect of this particular effort multiplier.

At this time, we produce the report for each effort multiplier at the same time. As a result, we assume implicitly that the influence of the other effort multipliers is estimated correctly by the COCOMO II-model and we therefore don't take into account the new knowledge we receive from the reports, namely that these factors are not always correctly estimated in relation with this company. In future research, we will investigate whether it is possible to use a 'step-wise regression' technique to incorporate this extra knowledge into the report for the effort multipliers.

The results obtained with these reports, namely the new values for the effort multipliers, can be used to calibrate the effort multipliers in the COCOMO II-model. In the case the model is used as an estimation tool, using the obtained values for calibrating the model would definitely improve the accuracy of the predictions, even with such a small amount of data. However, in our case, the model is mainly used as a productivity measurement instrument. We want to measure the productivity, identify where improvement is possible and finally measure this improvement. By calibrating the model, we will lose the initial benchmark and the new measurement results will no longer be comparable with the former obtained measurements. As such, it will be impossible to measure an improvement. Therefore, in this stage of the project, we will not yet use the obtained results to calibrate the effort multipliers.

Next to finding a way to incorporate prior knowledge, we also have to investigate how we can capture the productivity improvement with this report. An improvement in productivity can result in two effects. Firstly, an action leads to a shift in the rating of an effort multiplier. This improvement will not be visible in the report as the estimated effort by the COCOMO II-model will incorporate this improvement by giving another rate to the effort multiplier. However, a simple report with the frequency of each rating of the effort multiplier enables us to detect this shift in rating. Secondly, an improvement might lead to an effort reduction rather than a shift in the rating. This change will be visible in a change in the slope of the regression lines. However, when a company performs several actions on different effort multipliers, it is not clear to which action the measured improvement can be attributed. Indeed, suppose a company identified two actions  $A_1$  and  $A_2$  on two different effort multipliers  $EM_1$  and  $EM_2$  that lead to a global effort reduction of 10%. How much of this reduction can be assigned to action  $A_1$  and how much to action  $A_2$ ? The reduction in the actual effort will lead to a reduction in the Normalised Effort for both effort multipliers  $EM_1$  and  $EM_2$ . As a result, for both effort multipliers, there will be a change in the slopes. How much of the change we measure in the slope of  $EM_1$  is actually an effect of action  $A_1$  and how much is the result of the action on the other effort multiplier? In summary, the question remains how we can deduce to which action we can attribute the effect in productivity we measure. To solve this question we would need projects for which only action  $A_1$  was applied and a set of projects for which only  $A_2$  was applied. As management is aiming at maximal productivity improvement, the company is not willing to invest in such experimental set up and all projects will be allowed to benefit from all software process improvement actions.

## 6. Acknowledgements

This paper has been written as part of the ‘KBC-research chair’-project on ‘Managing efficiency aspects of software factory systems’ sponsored by KBC Bank en Verzekeringen.

## 7. References

- [1] Boehm, B.W. “Software Cost Estimation with COCOMO II”, Prentice Hall, New Jersey, 2000.
- [2] De Rore, L., Snoeck, M. and Dedene, G., “Cocomo II applied in a bank and insurance company: Experience Report”, Proceedings of the 3rd Software Measurement European Forum, May 10-12 2006, Rome, pp. 247-257.